CONNEXIONS™

The Interoperability Report

April 1990 Volume 4, No. 4

ConneXions—
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop, Inc., 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779.

ISSN 0894-5926

From the Editor

In our series *Components of OSI* we come to the OSI file service called FTAM (File Transfer Access and Management). FTAM corresponds roughly to the Internet FTP service, but offers greater functionality. In particular, FTAM has the ability to deal with complex data structures and different document types, and the FTAM standard addresses issues like record locking and file access control. The article is by Klaus Truoel from DFN/GMD who has been heavily involved in the development of FTAM in the International standardization community. In addition to giving a tutorial on FTAM, Klaus describes the important work of the international OSI workshops that define profiles from which standards can be implemented.

The specification of a new *Point-to-Point Protocol* (PPP) for serial lines was released in November of 1989 in the form of RFC 1134. This marked the culmination of 2 years of protocol development. A replacement for the old SLIP (Serial Line IP) protocol was first proposed at INTEROP 87. As is often the case in the Internet community, the decision reached was to start from scratch and address the wider issue of point-to-point circuits, rather than to simply "improve" (and document!) SLIP. Russ Hobby of UC Davis gives the history of this protocol development, and outlines the main features of PPP.

Plans are already well underway for INTEROP 90. As usual, there will be a large exhibit network where vendors will be demonstrating interoperable systems. A short description of these demos can be found on page 21. Also, on the same page, you'll find a brief statement from the RFC Editor, Jon Postel, on the format of RFCs.

Previous articles in *ConneXions* have talked about the *Internet Engineering Task Force* (IETF). [In particular see Volume 2, No. 10, October 1988]. We asked Greg Vaudreuil of NRI to give us an IETF update, in light of the recent explosive growth and addition of a steering group to this task force.

Bob Stine from SPARTA, with help from the Internet community at large, has compiled a document entitled *A Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices.* Recently published as RFC 1147, this document promises to be of great value to site administrators, network programmers, and network managers. Details about this important catalog can be found on page 25.

Also in this issue you'll find two book reviews, an announcement of the INTEROP Achievement Award, and a list of upcoming events.

Components of OSI: File Transfer, Access and Management (FTAM)

by Klaus Truoel, Deutsches Forschungsnetz

Introduction

The transfer of files across a network and the access to parts of a file is a very important application function which has been in use for many years. The current disadvantage is that there are many different protocols being implemented, e.g., the ARPANET FTP, the Blue Book protocol in the UK and a large variety of proprietary systems offered by vendors. Interworking between these systems is only possible in a very few cases where gateways are provided, usually offering only a reduced functionality as compared to the interconnected systems.

Typically, networking aims at worldwide interconnection through interworking between heterogeneous systems. The ISO FTAM base standard which is gradually coming into operation, aims at such a standardized and open interconnection of file transfer systems. This article gives a general introduction to the functions of the FTAM standard, their mapping into application functions—called functional standards—and a prospect of what the user may expect regarding the future use of standardized file transfer products.

FTAM—The standard

FTAM—*File Transfer, Access and Management*—is an Open Systems Interconnection (OSI) standard describing a service and its supporting protocol that allows a user to read or write files (or parts of them) in a remote filestore and to manage those files. This 5-part standard ISO 8571 [1] was completed at the end of 1987.

As an OSI standard FTAM fits completely into the 7-layer Basic Reference Model for Open Systems Interconnection (Figure 1). It is an Application Standard, describing the communication aspects of an application function in the Application Layer.

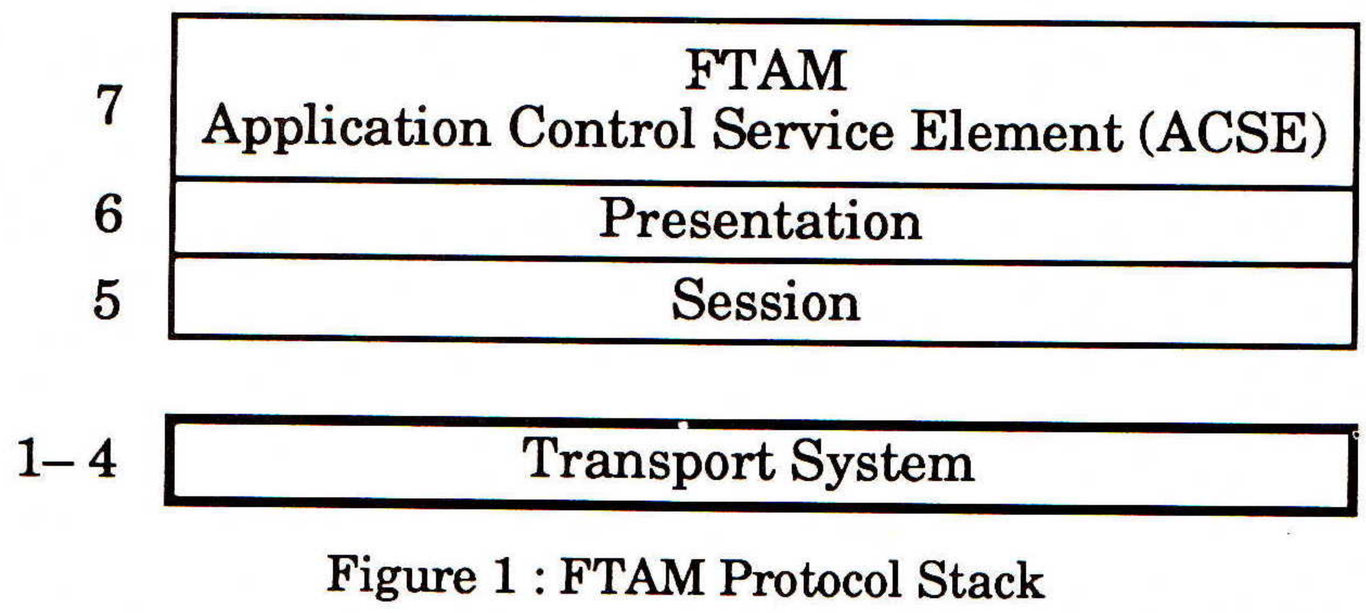


Figure 1 : FTAM Protocol Stack

Therefore, requirements to ACSE, Presentation and Session Services are also described. But the FTAM application function itself is independent of the underlying Transport System which is in use, the only requirement is a connection-mode transport service.

Virtual Filestore

Real files and real data management systems are implemented in a wide range of different styles in existing systems. In order to define a file transfer protocol for an OSI environment there is a need for a unified view on the concept of files, for a general file model (or a small set of different but standardized file models).

The hierarchical file model which is currently defined in the FTAM standard is called the *Virtual Filestore*. It describes files as a hierarchical structure, as a tree structure of nodes, each node containing some structural information, e.g., a name, and optionally a unit of user data, called a *Data Unit* (DU) (Figure 2).

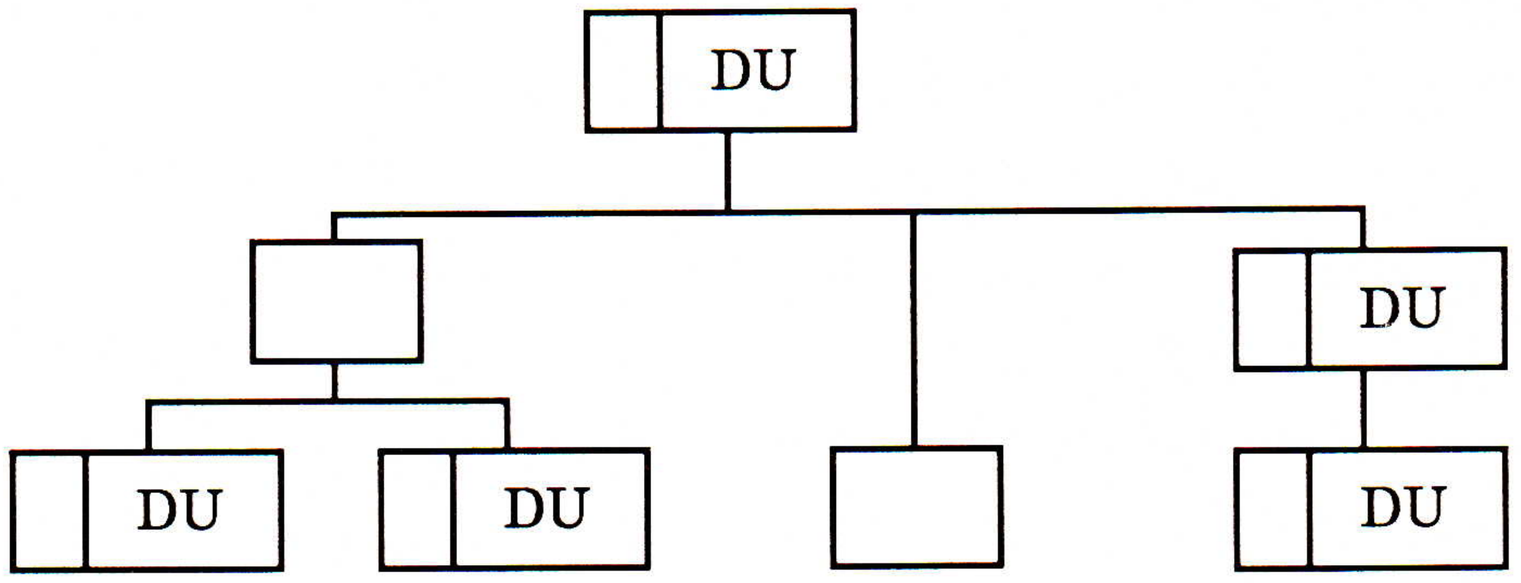


Figure 2 : FADU Access Structure

FADU

A node, together with all its subsequent nodes which are connected to it, is called a *File Access Data Unit* (FADU). These FADUs, either the full subtree of a FADU or only the root node of it, are the units in terms of which access to a Virtual File is achieved. They are addressable for transfer and for access.

The OSI view on the Filestore of a remote system is always that of its Virtual Filestore. The embedding of the Virtual Filestore into a system's Real Filestore with the corresponding mapping function is outside the scope of the FTAM standard (Figure 3). This mapping is the implementor's choice and also gives room for additional multilateral agreements between FTAM users.

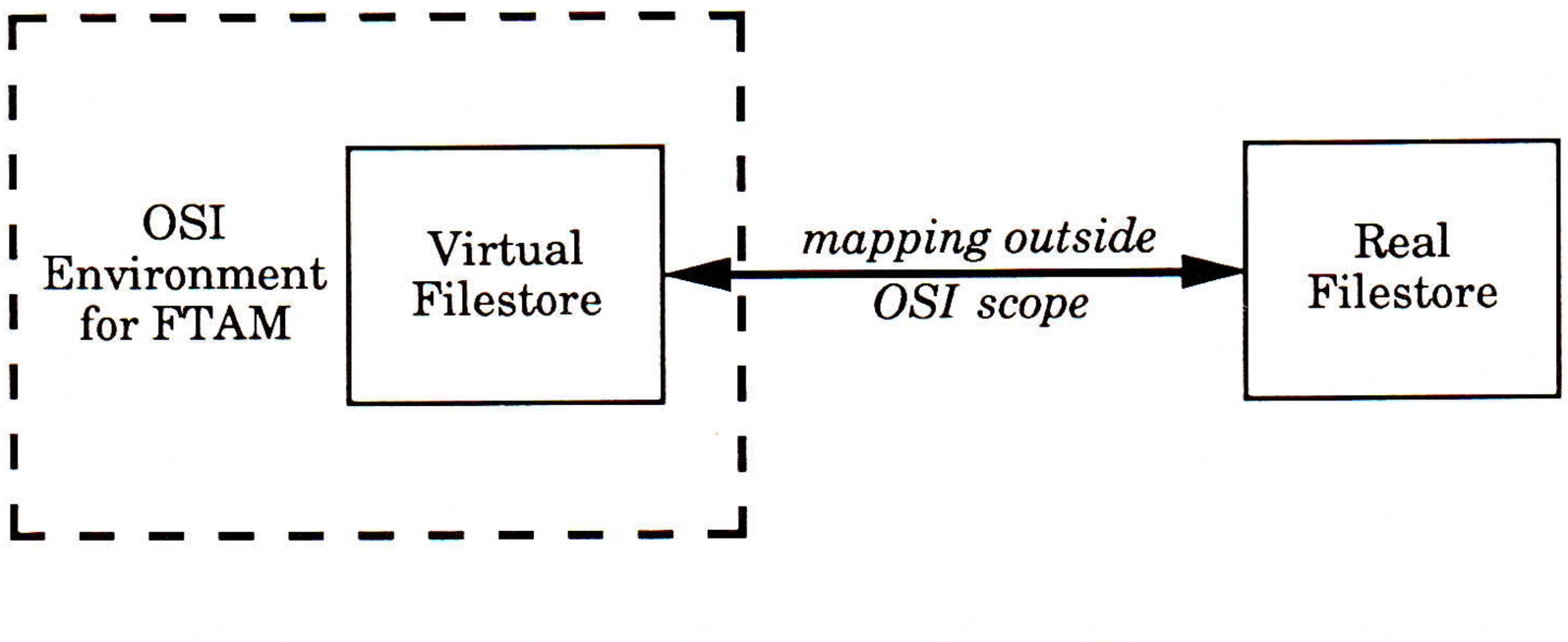


Figure 3: Virtual Filestore and Real Filestore

Associated with a file in the Virtual Filestore is a list of *file attributes*, describing characteristics of that file, e.g., file name, information on structure and contents, access constraints, access controls, security aspects, accounting information, etc.

Constraint sets

For a more convenient use of the file model and mapping onto the functionality of a real file system, the concept of *constraint sets*, giving another attribute for a file, defines several useful subsets of the general hierarchical structure.

continued on next page

File Transfer, Access and Management *(continued)*

Some of these constraint sets are (see also Figure 2):

- *Unstructured:* 1-level hierarchy of file data. A single data unit without a name and no subsequent child nodes appended to it.
- *Sequential flat:* 2-level hierarchy of file data. A root FADU without a data unit and a series of unnamed FADUs each with or without data units.
- *Ordered flat:* 2-level hierarchy. A root without a data unit and a series of named FADUs, each with or without data units.
- *Ordered flat with unique names:* 2-level hierarchy as ordered flat, but with uniquely named FADUs.

Document Types

A further step towards facilitating for real applications is the concept of *Document Types*. A Document Type describes a class of virtual files by specifying the same constraint set for all these files, the abstract syntax, i.e., the data structure of the Data Units, the transfer syntax, other structural constraints and semantics for the use of the Document Type. A Document Type is registered and identified by a worldwide unique *Object Identifier*, a structured integer, and it can therefore be uniquely referenced in an FTAM communication. Some Document Types are defined in the FTAM standard, others may be specified and registered according to user needs. A few examples of FTAM defined Document Types are:

- *FTAM-1:* Unstructured file of text data. Text means character sets identified by a parameter, e.g., ISO 646, ISO 8859, ISO 6937 character sets.
- *FTAM-2:* Sequential flat file of text data. Corresponds to a sequence of text FADUs.
- *FTAM-3:* Unstructured file of binary data.
- *FTAM-4:* Sequential flat file of binary data.

As already mentioned, a FADU and its associated Data Unit are the minimal portions of a virtual file in terms of which access is defined. The actual transfer of such a data unit—which might be quite a long piece of data—can be accomplished through subdivision into smaller pieces, called *Data Elements*. A sender may send a Data Unit piecewise, split up into several Data Elements, and the receiver can reassemble these Data Elements to build up the Data Unit. This is how additional semantics may be transferred for a Data Unit as illustrated in the following examples of mapping between real and virtual file structures.

Example 1: A UNIX text file of ISO 646 text, which is a data stream without record structure, is transferred as an FTAM-1 file, i.e., as one Data Unit. The subdivision into Data Elements for transfer is arbitrary and carries no semantics. A line-structure of that file is transmitted with the control characters Carriage Return, and Linefeed.

Example 2: A line- or record-oriented text file without control characters may be transferred as an FTAM-1 file, each Data Element carrying the additional semantics of representing a record. It may also be transferred as an FTAM-2 file, each Data Unit carrying exactly one record (and in this case the further subdivision into Data Elements either not applied or at least carrying no additional semantics.)

The method used depends on the creator of the virtual file and will be indicated via the appropriate file attributes, e.g., the Object Identifier of the document type for that file, including some parameters for the character set which is in use, the way Data Elements are handled, etc.

File service

The File Service and its supporting Protocol define two roles, that of an *Initiator*, initiating and controlling the FTAM association between two end systems, and that of a *Responder*, providing access to its Virtual Filestore. The FTAM communication is always asymmetrical. An Initiator system reads, writes, accesses or manages the Virtual Filestore of a Responder's system (Figure 4).

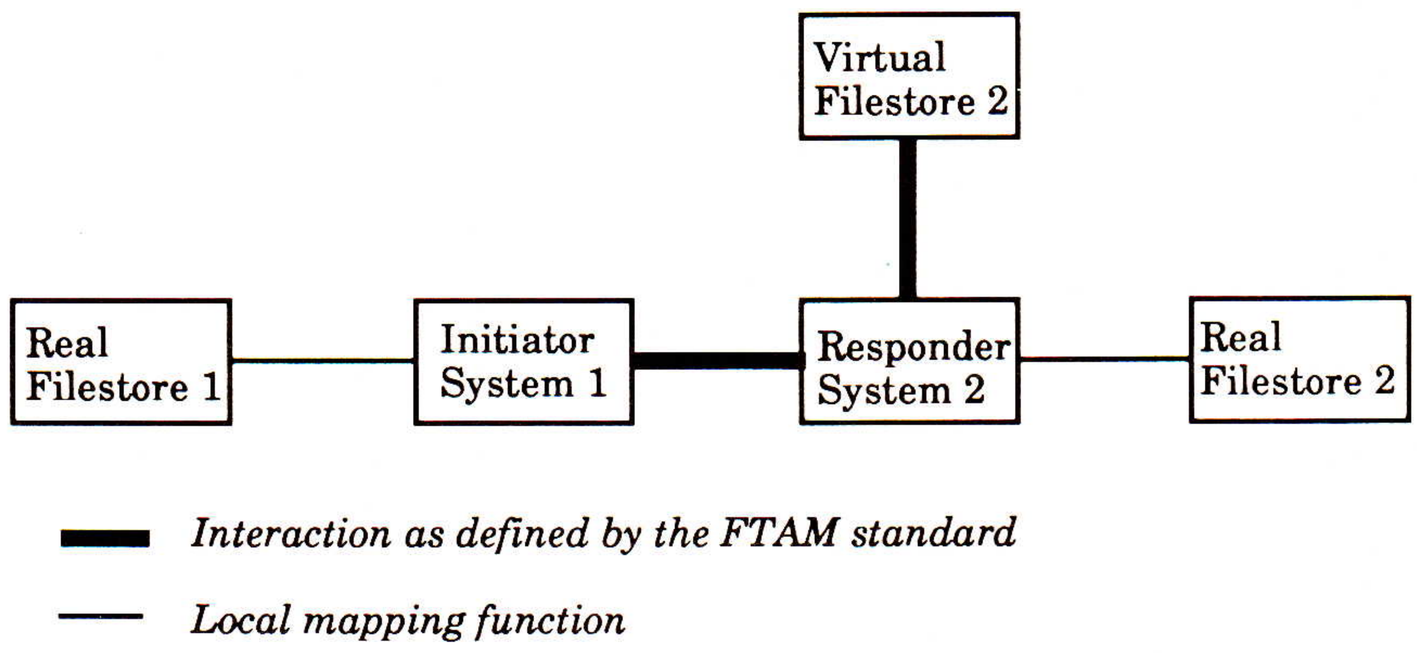


Figure 4: FTAM Service Model

Functional Units

FTAM specifies a large variety of services which cover the necessary functions for all kinds of access to and manipulation of virtual files. In order to help users select those functions which are needed for a specific FTAM association and negotiate them with the peer system, the FTAM services are grouped into *Functional Units*. Negotiation between Initiator and Responder at FTAM association establishment time is in terms of these Functional Units. A short overview of the defined Functional Units follows:

- *Kernel:* Basic functions of establishing and releasing an FTAM association, selecting and deselecting of a file for further processing.
- *Read:* Open and close a file, read the complete file or a specified FADU.
- *Write:* Open and close a file, write the complete file or insert, replace, extend a specified FADU.
- *File Access:* Locate a specific FADU for subsequent read, write or erase.

continued on next page

File Transfer, Access and Management *(continued)*

- *Limited File Management:* Create, delete files, read attributes of a file.
- *Enhanced File Management:* Change attributes of a file.
- *FADU Locking:* Specify access locks (concurrency locks) to FADUs of a file.
- *Recovery, Restart:* Recover from local errors or transfer and association errors by using the concept of checkpoints and dockets, the latter preserving all necessary information of the FTAM association in order to recover after a failure.

Regimes

Typically, an FTAM association is structured into phases, called *Regimes*, which are completely nested into each other, each regime being initiated for the exchange of the corresponding FTAM functions and being finished afterwards. The FTAM file service regimes and their related functions as defined by ISO 8571 are the following:

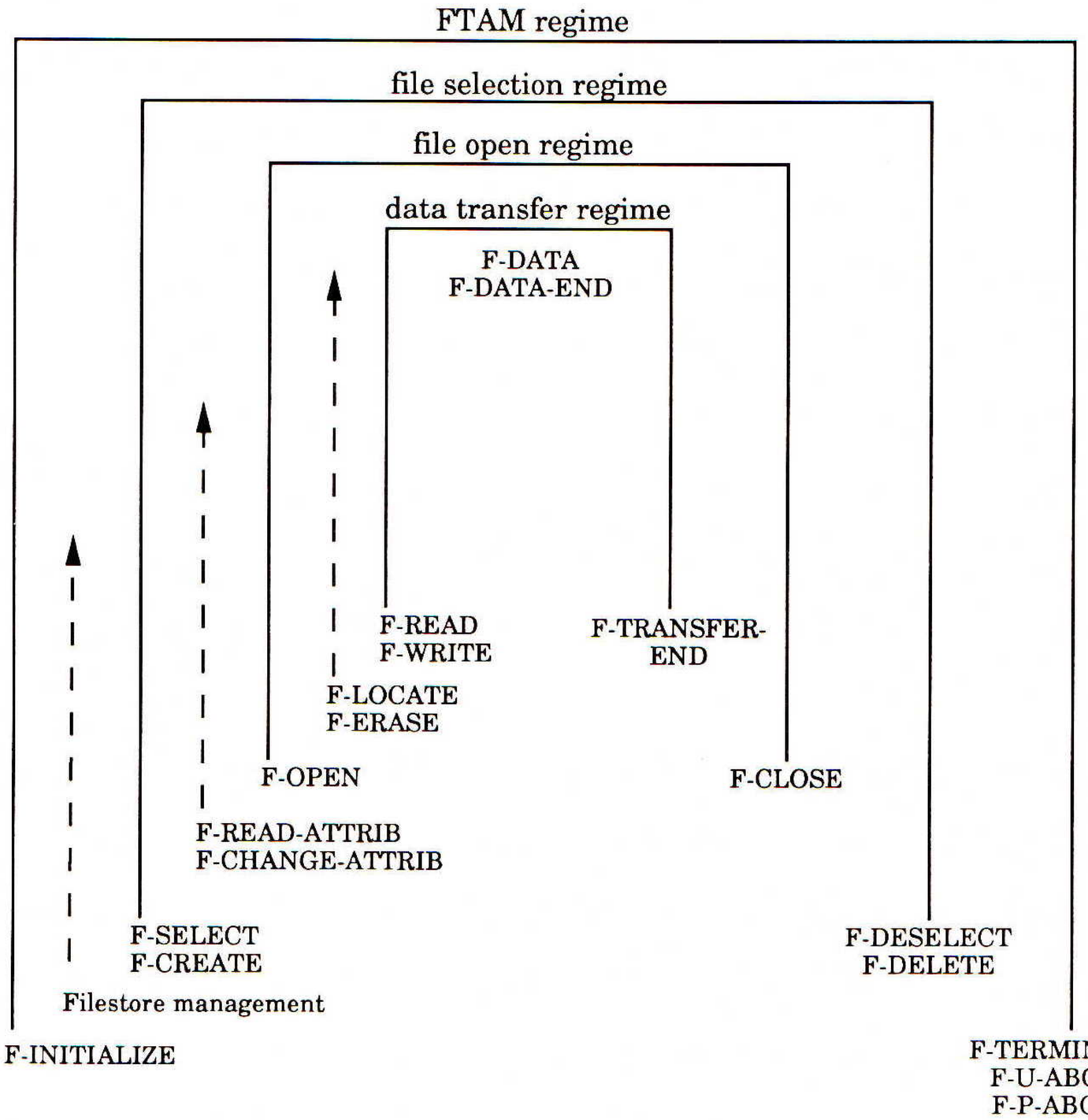


Figure 5: Model of Nested FTAM Regimes

Implementation profiles

Similar to most of the other OSI standards, FTAM defines a large variety of functions and options which cover the full range of possible access to remote files, but which need not be implemented in their entirety by all systems. That means interworking between different FTAM systems is possible only if both systems can dynamically negotiate and agree on a common subset of these functions and options being necessary for the current association.

The prerequisite for such a negotiable common subset is that a common set of functions is implemented on both systems. As a simple example, an Initiator system will not be able to access individual FADUs of a remote file if the Responder implementation supports only transfer of entire files. Access to a file of document type FTAM-1 containing Graphic String data will not be possible unless the Responder system supports these file types.

Therefore, the concept of Implementation Profiles was introduced in order to ensure interworking between different heterogeneous systems for similar application scenarios. An *Implementation Profile* or a *Functional Standard* is a specification, on the basis of one or more international standards, for a specific user function, stating how these base standards are to be implemented. It specifies subsets and option selections, and lists restrictions and additional agreements, in order to provide the intended function and to ensure interworking between different systems.

Regional OSI Workshops

The work on the definition of Implementation Profiles for FTAM is carried out mainly at three different places, in three groups of implementors and users, called the Regional OSI Workshops. These are:

- The *NIST Workshop for Implementors of OSI* at the National Institute of Standards and Technology in the US.
- The *European Workshop for Open Systems* (EWOS), located in Brussels.
- The *Asia-Oceania Workshop* (AOW) in the Japanese and Australian regions.

FTAM profile tree

In a very fruitful and successful cooperative work these three Regional Workshops have defined a set, more precisely a tree of FTAM profiles [3, 4, 5]. There is a strong need for a basic file-transfer Profile, which allows just for the transfer of entire files. No file access is needed for the basic Profile and the file access structure is only the simplest one. This Profile will fulfil the requirements of many user groups where only simple files are exchanged, in many cases of restricted length without the need for sophisticated restart and recovery mechanisms. This also allows for the implementation of the basic FTAM subset on small computers.

Other Profiles in the tree offer more functionality both in access capabilities and in access structure, ending up with the full power of the FTAM standard for general hierarchical files.

Figure 6 lists the six FTAM Profiles which are currently defined. The nomenclature differs slightly between the NIST OSI Workshop [3], the European Workshop (including the European Norms issued by CEN/CENELEC, [4]) and the Taxonomy of International Standardized Profiles [5]. However, the contents of these FTAM Profiles are fully harmonized and technically identical, so that worldwide interworking will be possible between implementations based on these Profiles.

continued on next page

File Transfer, Access and Management *(continued)*

	NIST OIW	EWOS	CEN/CENELEC	ISO/IEC ISP
File Transfer Simple File Transfer Positional File Transfer	 T 1.3 T 2.3	 A/111 A/112	 ENV 41204 ENV 41206	 AFT 11 AFT 12
File Access Positional File Access	 A 1.3	 A/122	 ENV 41207	 AFT 22
File Management	M 1.3	A/13	ENV 41205	AFT 3
Filestore Management	not yet defined			

Figure 6 : FTAM Implementation Profiles

It should be noted that the two Profiles for file transfer are upward compatible.

Simple File Transfer

The *Simple File Transfer Profile* covers transfer of files between an initiating end system and the Virtual Filestore of a responding end system. Transfer of files is supported only for complete files with an unstructured constraint set (Document Types FTAM-1 and FTAM-3, another type for a file-directory is added as an option). No access to parts of a file is provided. It therefore defines the ability to:

- Read a complete file of one Data Unit.
- Write (replace, extend) to that Data Unit.
- Optionally create and delete a file.
- Optionally read the attributes of a file.
- Optionally include the restart and recovery functions.

Positional File Transfer

The *Positional File Transfer Profile* offers transfer functionality only, but in this Profile both unstructured and flat files are supported. Flat files means that an individual File Access Data Unit read or written The document types FTAM-l, FTAM-2 and FTAM-3 for text and binary data are supported together with some additionally defined optional Document Types, e.g., a mixed data-type file with or without keys, a simple random-access file, a file-directory file, etc. This Profile is upward compatible with the simple Profile.

Positional File Access

The Positional File Access Profile provides the functions of repeated transfer of and access to a file or its individual FADUs. The allowed constraint sets are unstructured and flat, i.e., a 1-level or 2-level hierarchy of binary and of character data. It defines the ability to:

- Read a complete file or FADUs which are identified by node name, node number or by their relative position in the file.
- Write (replace, extend, insert) to a file or a FADU.
- Locate the position pointer to a FADU, erase a FADU.
- Optionally create and delete a file.
- Optionally read the attributes of a file.
- Optionally include the functions of restart, recovery and of locking FADUs.

File Management

The functions of creating a file, deleting a file, reading attributes of a file, and changing attributes of a file are described as mandatory in the *File Management Profile*. This Profile is only to be implemented in conjunction with one or more of the Transfer or Access Profiles.

Filestore Management

An addendum to ISO 8571 is currently under development by ISO. It defines the services and a protocol for an FTAM Initiator to manage file-directories in the Responder's Virtual Filestore, e.g., create, delete, copy file-directories, select groups of files, move files in a file-directory. Since this ISO base standard is not yet stable, corresponding work on a *Filestore Management Profile* will not be started by the OSI Workshops until mid-1990.

International Standardized Profiles

ISO as a body for base standardization recognized the importance of the definition of Functional Standards or Implementation Profiles only a few years ago. Only by that means can the applicability of OSI standards be promoted and at least a certain guarantee of interworking be given. Therefore, a Special Group on Functional Standardization, the Working Group on Taxonomy of ISO/IEC JTC 1 defined the concept of *International Standardized Profiles* (ISPs).

The general objective of an ISP is again to identify base standards including the specification of selected subsets, options and parameters in order to accomplish an identified function for users. And this for real international use, rubber-stamped by ISO. An ISP should leave as few options as possible, be unambiguously implementable and therefore ensure interworking for the relevant application scenario.

A taxonomy of Profiles [2] gives a complete list and classification of the ISPs which are currently included for future development. Six classes of ISPs are defined:

T: *Transport Profiles* providing connection-mode Transport Service.

U: *Transport Profiles* providing connectionless-mode Transport Service.

R: *Relay Profiles.*

A: *Application Profiles* requiring connection-mode Transport Service.

B: *Application Profiles* requiring connectionless-mode Transport Service.

F: *Interchange Format* and representation Profiles, e.g., for ODA.

For FTAM the substructure of the Application Profiles is as listed in Figure 6 above. This structure and the technical contents are fully equivalent to those of the FTAM Profiles of the Regional Workshops.

A very important part of each ISP is the *ISP Implementation Conformance Statement Requirements List* (ISPICS Requirements List, IPRL), providing in a tabular form a summary of all functions and options of the respective base standards, and all selected options, agreements and restrictions of the Profile. Therefore, such an *IPRL* gives a clear and identical view, both to users and suppliers, of the properties and the behaviour of a product with respect to an ISP.

continued on next page

File Transfer, Access and Management *(continued)*

Approval of ISPs will be done using a simplified ISO procedure. Stable Profile specifications which are agreed upon among the three Workshops at NIST OIW, EWOS, and AOW, and sufficiently harmonized and aligned between these workshops, will pass an accelerated 3-month ballot procedure with the ISO member bodies prior to being published as ISPs. The main task of preparing and specifying an ISP will stay—as it is already currently in place—with the groups of vendors, implementors and users, organized in the three Regional Workshops. Seven parts of an FTAM ISP [5] are currently under ISO/IEC ballot.

Conformance tests

The main characteristic of software systems interworking in an open environment is their heterogeneous nature. They are implemented by different programmers for different operating systems. The protocol standards specify a very complex external communication behavior of these systems. It is therefore a very important prerequisite for a successful introduction of OSI systems to have tools available which support testing, that ensure correct interworking capability with respect to the base OSI standards and the Profile specifications.

These test tools are important for the implementor in order to give him a continuously growing confidence, while building the product, that the development follows the right track. They are also a fundamental user requirement. Customers would not purchase OSI products aiming at interworking with other systems, unless they could get some guarantee that the protocol specifications were fulfilled.

Tests of this kind are important as interworking tests between arbitrary systems with a set of predefined test cases and also as tests against an officially accredited and verified test system, which executes a series of test suites for as many protocol paths as possible.

The specification of test cases and test suites for OSI applications, which allow for a fairly complete test of a system's protocol behavior, is currently under work at many different places, at ISO, in research projects and vendor groups. It is the goal to build up test centers which are open for users and vendors and which will in some cases be authorized for official conformance tests, giving test and conformance certificates to products having successfully passed such a test.

Work of that kind including the establishment of test centers is currently being done at COS, ITI and OSINET in the US and in the European project CTS (Conformance Test Services), by SPAG in Brussels, and by the German FTZ in Wiesbaden, among others.

The users' view

Standardization and Functional Standardization are currently well advanced for FTAM. Profiles are specified by the Workshops and by ISO/IEC. Now it is up to the user to step forward with his requirements for an OSI-style file transfer, based on the available Profile specifications.

Large user groups such as the US Government (US GOSIP), the UK Government (UK GOSIP), COS, MAP and TOP, the OSINET project, and the European academic networking projects RARE and COSINE have already defined their requirements for FTAM services which are fully based on these Profiles.

FTAM implementations for nearly all operating systems are currently available or at least nearing their completion. Gateways to existing non-OSI file transfer systems, e.g., ARPA FTP, are under development. Test centers and interoperability projects are already operational. Therefore, the door to an FTAM future of file transfer is already open.

References

[1] ISO 8571: 1988 (E) "Information Processing Systems—Open Systems Interconnection—File Transfer, Access and Management."

Part 1: General Introduction
Part 2: Virtual Filestore Definition
Part 3: File Service Definition
Part 4: File Protocol Specification
Part 5: Protocol Implementation Conformance Statement Proforma (1990).

[2] ISO/IEC TR 10000: 1990 (E), "Information Technology—Framework and Taxonomy of International Standardized Profiles."

Part 1: Framework
Part 2: Taxonomy of Profiles

[3] "Stable Implementation Agreements for Open Systems Interconnection Protocols," Version 3, Edition 1, December 1989, NIST Special Publication 500-177.

[4] CEN/CENELEC European pre-Norms:

ENV 41204: 1990 Simple File Transfer (Unstructured)
ENV 41205: 1989 File Management
ENV 41206: 1990 Positional File Transfer (Flat)
ENV 41207: 1990 Positional File Access (Flat)

[5] ISO/IEC DISP AFTnn, "Information Technology—Draft International Standardized Profiles, AFTnn—File Transfer, Access and Management."

Part 1: Specification of ACSE, Presentation and Session Protocols for the use by FTAM (Jan 1990).
Part 2: Definition of Document Types, Constraint Sets and Syntaxes (Jan 1990).
Part 2: Definition of Document Types, Constraint Sets and Syntaxes. Addendum 1: Additional Definitions (April 1990).
Part 3: AFT11 Simple File Transfer Service, Unstructured (Jan 1990).
Part 4: AFT12: Positional File Transfer Service, Flat (April 1990).
Part 5: AFT22 Positional File Access Service, Flat (April 1990).
Part 6: AFT3 File Management Service (April 1990).

KLAUS TRUOEL received his doctor's degree in Mathematics from Bonn University in 1968. For the past 10 years he has been working in the field of Telecommunications and OSI protocols at the German research company GMD and for the German Research Network DFN. He was and still is involved in the specification of Implementation Profiles for FTAM as chairman of the NIST OIW FTAM group, editor of the FTAM ISPs, and member of the FTAM groups of EWOS and AOW.

Book Reviews

In the book, *UNIX Network Programming* by W. Richard Stevens, Prentice Hall, ISBN 0-13-949876-1, the author ambitiously attempts to present the concepts and details needed to develop sophisticated networking applications for UNIX-based systems. The book covers both the 4.3 BSD and System V versions of UNIX, and it includes 15,000 lines of source code ranging in complexity from a simple client-server program using pipes, to the BSD UNIX remote login utilities *rlogin* and *rlogind*. Although the focus of the book will be welcomed by UNIX programmers, its presentation of the material is unlikely to please either beginning or experienced UNIX programmers.

Organization

The book is divided into roughly four parts. The first part introduces concepts and reviews the history of the many versions of the UNIX operating system. In addition, it provides a brief summary of such basic UNIX concepts as processes, process groups, the file system, file protection, signals, as well as related system calls. The remainder of the first part covers intra-machine interprocess communication, focusing on pipes, FIFOs, streams, shared memory, and System V's message queues and semaphores. Each topic includes an example program that demonstrates how a program actually uses a particular facility.

Networking

The second part of the book covers such general networking concepts as layering, local and wide area networks, gateways, the client–server model, and routing. It also presents various protocols that have been used in and around UNIX systems, including the TCP/IP protocols, the Xerox *XNS* protocols, *UNIX-to-UNIX Copy* (UUCP), as well as *NetBIOS*, and IBM's *Systems Network Architecture* (SNA). At this point the focus of the book narrows considerably, with the remaining five hundred pages of text focusing almost exclusively on the TCP/IP and XNS protocol suites.

Sockets and TLI

The third part of the book gives detailed descriptions of BSD *sockets* and System V's *Transport Layer Interface* (TLI), and covers the UNIX library routines for such functions as converting between user-friendly machine names and low-level addresses or converting between host and network byte ordering. In addition, the author develops a set of routines for dynamically estimating the round trip time and presents an application that uses acknowledgements, timeouts, and retransmissions to provide reliable service using the unreliable UDP service.

Applications

The last part of the book covers applications and security. One chapter shows the source code for the authentication steps used in BSD's *rsh* and *rexec* family of applications and gives an overview of MIT's *Kerberos* system. Another six chapters provide complete, substantial application programs that are both useful in their own right and tie together the concepts and details of the previous chapters. A chapter on *ping* shows a program for sending ICMP echo requests, along with its analog for the XNS environment. Another chapter presents a client implementation of the Trivial File Transfer Protocol (TFTP), while another describes the BSD line printer spooler programs *lpr* and *lpd*, giving a sample *lpr* client program capable of sending print jobs to a remote machine.

The book continues with a presentation of the source code for the BSD remote command programs *rcmd*, *rshd*, *rexec*, and *rexecd*, and it devotes a chapter to the *rlogin* remote terminal login routines, describing pseudo-terminals, flow control, and the need for control terminals. The remaining chapters cover remote tape drive access, performance issues, and remote procedure calls.

Shortcomings

While the book is filled with information, its presentation of material leads to immediate difficulties. Undergraduate and graduate students as well as computer professionals will be frustrated in its failure to clearly explain concepts before diving into the details, rendering the book unattractive as a learning text. For example, while describing UNIX process ids, the author describes the real, effective, user, and group ids of a process, but gives the system calls used to obtain such ids without ever explaining (in abstract terms) what they are used for (e.g., to control access to resources). As another example, when describing the UNIX *sockaddr* data structure, the author dives into the details of what the fields contain for each of the various address types without first stepping back and explaining the purpose of structure itself (e.g., protocol-independent, transport-level addressing). Finally, the section describing reserved ports specifies which port numbers are reserved and how to allocate them, but fails to remind the reader what reserved ports are actually used for. While these examples may seem trivial, similar examples appear throughout the text.

Likewise, experienced UNIX programmers will be disappointed in the book's weakness as a reference text. When discussing library procedures and system calls, the book gives examples of how a program would use a particular routine, but neglects to clearly articulate what the routine actually does, or what its arguments are. Indeed, explanations tend to be operational (e.g., set field X to 54 and then call Y) rather than explanatory in nature, and programmers will need to consult the UNIX manual pages for an explanation of what a particular routine does. While there is no substitute for the actual manual pages, of course, the book omits too many details to pass as a reference text. For example, some of the sample programs use types and symbols (presumably) defined in header files, but not included in the text. To fully understand the examples, access to the actual manual pages is a must.

Good example programs

In balance, the book's strongest contributions are its programs. The large number of sample programs demonstrate the usage of UNIX system calls and library procedures, and the book clearly highlights the differences between BSD and System V UNIX. The example programs are especially useful because they also cover obscure, poorly documented features such as sending file descriptors across sockets, or the setting of various socket options.

Disappointing

In summary, *UNIX Network Programming* is a disappointing addition to the libraries of UNIX network programmers. Although it has assembled a wealth of information into a single book, and it includes large programs that demonstrate many of the advanced UNIX routines and system calls, most readers will find the book either lacking in detail, or difficult to follow. *—Thomas Narten*

continued on next page

Book Reviews *(continued)*

Handbook of COMPUTER-COMMUNICATIONS Standards Volume 3: The TCP/IP Protocol Suite, Second Edition by William Stallings, Ph.D. with (not enough) help from Paul Mockapetris, Sue McLeod, Tony Michel, Craig Partridge, and Keith McCloghrie. Published by Howard W. Sams & Company, ISBN 0-672-22696-0.

Architecture and Core Protocols

If you are looking for a textbook case of literary *schizophrenia*, then this book is for you. The first half discusses the architecture and core protocols of the Internet suite (TCP and IP); whilst the second half of the book discusses the most common application protocols used in the Internet suite. I found the first half, written by Stallings, to be largely incoherent. In contrast, the second half, written by various Internet experts, was quite good.

Terminology

When describing a technology it is important to use appropriate terminology. Unfortunately, Stallings sees fit to cast the Internet protocol suite in terms of the OSI model and notation. And, just as those who have been trying to retrofit X.25 into the OSI model for the last decade, the final product just doesn't make sense. For example, with only one exception, Stallings never uses Internet terms when describing Internet concepts: instead of *packets*, he uses *PDUs*; instead of *fragmentation*, he uses *segmentation*, and so on. (The one exception is that Stalling does use the terms *host* and *gateway* rather than the OSI terms *end-system* and *intermediate-system*.)

The problem with this approach is that someone who first reads Stallings' Second Edition, and then tries to converse with people with first-hand experience with the Internet technology, will speak with a very "bad accent," leading to confusion and inaccuracy. So, if you are an OSI person who wants an explanation of Internet technology cast in your native tongue, then this approach is suitable. Otherwise, if you are trying to learn about and then *use* Internet technology, then this is an ill-suited text for your purpose.

Application Protocols

Fortunately, the second half of the book contains a good look at many of the well-engineered application protocols which have made the Internet suite of technology the de facto standard for delivering services to users: FTP (file), SMTP (mail), Telnet (terminal), and network management. Of these four, my favorites are the discussion on electronic mail by Mockapetris and the introduction to network management by Partridge and McCloghrie.

The electronic mail discussion does a thorough job of discussing the Simple Mail Transfer Protocol (SMTP, RFC 821), but is somewhat terse on the actual format protocol (RFC 822) which defines the messages that are exchanged using SMTP. Nonetheless Mockapetris is detailed in describing the transfer protocol aspects with particular emphasis on the use of reply codes (used quite heavily in both mail and file transfer).

The network management discussion takes a more general focus, introducing both the framework and protocols. For example, the text describes "Case Diagrams" which are an important tool used to graphically describe the relationship between managed objects in a real implementation. Further, their discussion on the competition between SNMP and CMOT (a "RISC versus CISC" debate) is easily the most non-partisan description occurring in print as of this writing.

Missing applications

My major complaint with the second half of the book is the number of interesting application protocols and services that were left out. For example, who better than Mockapetris to write a chapter on the Domain Name System? In fact, this criticism is applicable towards every book on the Internet suite of protocols currently in print, including Comer's *Internetworking with TCP/IP*, is that no one spends enough time discussing the application protocols and services. In particular, there is little, if any discussion, on the really useful protocols that are not core, e.g., Sun's NFS, X Windows, the Post Office Protocol, and so on.

In summary

Although the preface to the 2nd edition notes significant reworking, many of the fundamental problems of the 1st edition were never addressed: Stallings is, by and large, an outsider trying to describe the TCP/IP technology using terminology and models that never quite fit: his explanation of TCP and IP are akin to fitting square pegs in round holes. It just never meshes.

My recommendation: if you want a good explanation of the fundamentals of the TCP and the IP, then get a copy of Comer's book, which provides much more insight than Stallings' Second Edition on this topic. However, if you want knowledgeable explanations of some of the key Internet applications, then this book may be for you—providing you are willing to write off the first half of the book as a sunk cost.

—Marshall T. Rose

Stay informed with *ConneXions!*

ConneXions covers a wide variety of topics in the field of computer communications. Free index pages of back issues are available on request. We are also working to develop a *Subject Index* which will make it even easier to find particular articles. From time to time we produce *Special Issues* on particular topics. Special issues include: *Protocol Testing* (August 1988), *Subnets* (January 1989), *Network Management* (March 1989), and *Internet Routing* (August 1989). A continuing series of articles entitled *Components of OSI* explains the emerging OSI standards. Articles to date include:

ISDN	April	1989
The X.400 Message Handling System	May	1989
The X.500 Directory Services	June	1989
The Transport Layer	July	1989
Routing overview	August	1989
IS-IS Intra-Domain Routing	August	1989
ES-IS Routing	August	1989
The Session Service	September	1989
CLNP	October	1989
The Presentation Layer	November	1989
A taxonomy of the players	December	1989
The Application Layer Structure	January	1990
FTAM	April	1990

All back issues of *ConneXions* are available for $15 each. You may also purchase complete volumes (1987, 1988 and 1989) in binders at a special price of $100 per volume. Empty binders are $5 each.

Let us know if you change your address so we can continue to send you *ConneXions* every month.

—Ole

The Point-to-Point Protocol (PPP)
A new proposed standard Serial Line Protocol

by Russ Hobby, University of California, Davis

Introduction

Point-to-point circuits in the form of asynchronous and synchronous lines have long been the mainstay for data communications. Even with the growing popularity of local area networks, wide area connections are still made using various point-to-point circuits. Over time the data link levels used have settled to a few standards. However, the method of use of the data link by network layer protocols, such as IP, has been largely implementation dependent. In the IP world, the defacto standard *Serial Line IP* (SLIP) [6, 7] protocol has served admirably in this area. However, since SLIP defines only the encapsulation, links were manually connected and configured. SLIP is also only defined for asynchronous links. What was needed was a protocol that could control the link and have the ability to configure the host appropriately. The *Point-to-Point Protocol* (PPP) addresses these problems.

History

A SLIP replacement was first discussed at an INTEROP® 87 Birds Of a Feather session. However, it wasn't until the *Internet Engineering Task Force* (IETF) formed the Point-to-Point Protocol Working Group in October of 1988 that real work on the new protocol was begun. There were two main camps in the working group. Those that needed a simple, efficient protocol for low speed links (know in the group as SLIPPERs) and those that wanted a standard for high speed, router-to-router links (known as SYNCHERs). It was decided to see if a single protocol could be designed to meet the needs of both groups since most of the needs for the two groups were common. There were three goals for the new protocol:

- A standard method of encapsulating datagrams over serial links for multiple network layer protocols.
- An extensible Link Control Protocol (LCP) to establish, configure, and test the data-link connection.
- A family of Network Control Protocols (NCP) for establishing and configuring different network-layer protocols.

In October of 1989 there were two independent implementations done of PPP to test the protocol for errors. The following month the working group produced a base document [1] defining PPP. The working group has continued work by producing a second document defining initial negotiated options [4] for the link and IP protocol, as well as a simple authentication protocol and a method for link testing while the link is operating.

PPP Applications

There are two main target applications for PPP. The first is router-to-router connections over serial links. Currently most router vendors provide serial link connections, but since each vendor has their own serial link protocol, both ends of the link must have the same vendor's equipment. PPP has created a standard so that different vendors' equipment can be on each end of a link. The major router vendors have participated in the definition of PPP and have committed to its implementation.

The second application is connection of host computers to a network over serial links. This can be accomplished either with dedicated links, or dial-up connections.

PPP has full connection and configuration capabilities, thus the entire connection phase can be automated. A PPP server, similar to a terminal server, can tell a connecting host all the configuration information, thus no special configuration of the host is necessary prior to connection. Several terminal server and workstation vendors participated in the definition of PPP.

Physical Layer

PPP is capable of operating across virtually any DTE/DCE interface. The most common examples of interfaces are EIA RS-232-C, EIA RS-422, EIA RS-423 and CCITT V.35. The only absolute requirement for PPP is that a *duplex circuit* be provided. While control signals are not required for the use of PPP, it should be recognized that they do add functionality and it is recommended that they be fully utilized when they are available from the interface.

Data Link Layer

PPP specifies for synchronous and asynchronous circuits that ISO 3309-1979 [2] as modified by ISO 3309:1984/PDAD1 [3] be used. The framing specified by this document is more commonly known as *HDLC* and the 1984 addendum defines HDLC for asynchronous circuits. HDLC framing has the advantages that it is an international standard and framing is often done in hardware. PPP describes the values to be used in the standard HDLC fields and adds a protocol field to indicate for which protocol the data is destined. There are three groups of protocols, link control, network-layer protocol control, and network-layer protocol data.

Bringing up a link

Before a link is considered to be ready for use by network-layer protocols, a specific sequence of events must happen. These are:

Phase 1: *Link Establishment and Configuration Negotiation*—In this phase, Link Control Packets are exchanged and link configuration options are negotiated. Once options are agreed upon, the link is OPEN, but not necessarily READY for network-layer protocols to be started.

Phase 2: *Authentication*—In this optional phase, each end of the link authenticates itself with the remote end using authentication methods agreed to in Phase 1. If no authentication is required, this phase is skipped. PPP currently defines a simple user/password authentication protocol. Development of other protocols is encouraged.

Phase 3: *Link Quality Determination*—If desired the link may be tested at this point to determine if the quality of the link is sufficient for operation. PPP does not specify the policy for determining link quality but does provide low level tools, such as echo request and reply, link quality monitoring for testing the line and checking for looped-back links. If no testing is required, this phase may be skipped.

Phase 4: *Link READY*—network-layer protocols may be separately configured, and may be brought up and taken down at any time. If the Link Control Protocol takes the link down, it informs the network-layer protocols so that they may take appropriate action.

Link Control Protocol

The *Link Control Protocol* (LCP) provides a method of establishing, configuring, maintaining and terminating the Point-to-Point connection. LCP handles configuration of the link itself, it does not handle configuration of individual network layer protocols.

continued on next page

The Point-to-Point Protocol (PPP) *(continued)*

All configuration parameters which are independent of particular network layer protocols are configured by LCP. LCP has packet types to provide link option negotiation, link up/down control, and link testing. Options that can be negotiated cover areas such as maximum receive unit, async control character mapping, authentication methods, encryption methods, and link quality monitor parameters. Since LCP was designed to be extensible, other options may be added in the future. All configuration options are assumed to be at default values until configuration exchange is completed.

Before any other protocol packets may be exchanged, LCP must first open the connection through an exchange of configure packets. This exchange is completed once a configure ack packet has been both sent and received indicating that both ends have agreed on the negotiated options. Any non-LCP packets received before this exchange is complete are silently discarded.

LCP provides the ability to do link testing, loop-back detection and link quality monitoring before and while the link is operating. PPP provides the tools for these functions but policy on when to take the link down or bring it back up is implementation dependent.

Link testing

Simple testing and loop-back detection can be accomplished using the LCP *Echo-Request* and *Echo-Reply* packets. These packets contain two important fields, the Magic-Number and Data fields. The *Data* field may contain any data the sender of the Echo-Request may wish to include.

The *Magic-Number* field contains a number that has been selected that will be unique to the device sending the packet. It is recommended that the Magic-Number be chosen in the most random manner possible in order to guarantee with very high probability that an implementation will arrive at a unique number. Suggested sources of uniqueness include machine serial numbers, other network hardware addresses, time-of-day clocks, etc. Once a Magic-Number has been selected for a PPP connection, that same number must be used during the life of the connection. When sending either an Echo-Request or an Echo-Reply packet the sender will put its own Magic-Number into the Magic-Number field.

As the names imply, when an Echo-Request packet is sent, the remote end is obligated to return an Echo-Reply packet. Any data received in an Echo-Request packet is copied into the Echo-Reply before sending. The receiver of an Echo-Request packet should inspect the Magic-Number field. If the Echo-Request contains the receiver's own Magic-Number, the receiver is seeing its own Echo-Request packet and the link is in loop-back. Upon receipt of the Echo-Reply, the requesting end can compare the data for accuracy and loss.

Link monitoring

LCP provides a method to determine the number of packets and octets lost during operation of a PPP link. The method is based on each end counting the number of packets and octets sent and received on the link from that end's point of view and sending this information to the other side to be compared with the counts there. The trick is the synchronization of this information.

Synchronization is accomplished by using a *Link Quality Monitoring* (LQM) packet as a checkpoint. The number of packets and octets transmitted are maintained in counters (counters are never reset) and when a LQM packet is sent the values of these counters are included. When the receiver gets the LQM packet, the values of the senders counters are compared with the received packet and octet counters, thus determining the number lost. To determine the loss rate, the number of packets lost over an interval must be determined. The interval used is between two LQM packet check points.

For example, let's compute the packet loss during the last interval based on the data:

	At Previous LQM Packet	At Current LQM Packet
Tx-packet-counter	6	15
Rx-packet-counter	5	12

It can be seen that the remote end sent 9 (15-6) packets and the local end received 7 (12-5) packets. Thus the packet lose for that interval was 2 packets. The same calculation can be done for octets. If an LQM packet is lost, the calculations are still valid, but the interval will be longer.

Each direction of the link is tested independently using LQM Packets. In addition to the counter values, the LQM Packet also includes return information so that both ends can know the link quality in both directions of the link.

IP Control Protocol

The *IP Control Protocol* (IPCP) is the Network Control Protocol for IP and is responsible for configuring, enabling and disabling the IP protocol on both ends of the point-to-point link. The IP Control Protocol state sequence for options negotiation is the same as the Link Control Protocol allowing for the possibility of reuse of code. Options currently defined for negotiation by IPCP include IP addresses and compression type.

IP Address Option

The IPCP Address option allows devices to verify or assign IP addresses with the remote end. This option contains two important fields that indicate *My-IP-Address* and *Your-IP-Address*. Unknown addresses are filled with zeros. Thus the option can be used for each end to tell the other its IP address. A device such as a router may want to reject a remote address if it can not route for that address.

If a device does not have an IP address, such as on a dial-up connection, zeros will be put in both fields thus requesting the other end to assign both My-IP-Address and Your-IP-Address.

Compression

Currently the only compression method that may be negotiated is Van Jacobson's Compressed TCP/IP. [5]

IP Data Packets

IP Data Packets are encapsulated and sent in PPP packets where the protocol field indicates DoD Internet Protocol. Exactly one Internet Protocol packet is encapsulated in the information field of PPP data link layer frames.

Implementations

Many vendors of routers, terminal servers and computers have committed to implementing PPP for their products. A few have now demonstrated their implementations.

continued on next page

The Point-to-Point Protocol (PPP) *(continued)*

There are two publicly available implementations of PPP. One, developed by Drew Perkins at CMU, for UNIX 4.3BSD. The other, for PCs and based on Phil Karn's KA9Q code, was developed by Katie Stevens at UC Davis.

Future work

There are many areas for future work on PPP. Currently only use of the IP protocol has been defined on PPP. To make use of PPP for other protocols, the control protocol and rules for the data encapsulation need to be defined. Work is being done to define how to use PPP with bridging, OSI, DECNet and XNS. The authentication method currently defined is quite simple. A stronger authentication method needs to be defined. Also, while there is a means to negotiate encryption methods, no encryption methods have been defined.

References

[1] Perkins, D., "The Point-to-Point Protocol: A Proposal for Multi-Protocol Transmission of Datagrams Over Point-to-Point Links," RFC 1134, November 1989.

[2] International Organization for Standardization, "Data Communication—High-level Data Link Control Procedures—Frame Structure," ISO Standard 3309-1979, 1979.

[3] International Organization for Standardization, "Information Processing Systems—Data Communication—High-level Data Link Control Procedures—Frame Structure—Addendum 1: Start/stop Transmission," Proposed Draft International Standard ISO 3309:1983/PDAD1, 1984.

[4] Perkins, D., "The Point-to-Point Protocol (PPP) Initial Configuration Options," RFC (xxxx), (To be published).

[5] Jacobson, V., "Compressing TCP/IP Headers for Low-Speed Serial Links," RFC 1144, March 1990.

[6] Romkey, J., "SLIP: Serial Line IP," *ConneXions*, Volume 2, No. 5, May 1988.

[7] Romkey, J. L., "A nonstandard for transmission of IP datagrams over serial lines: SLIP," RFC 1055, June 1988.

RUSS HOBBY received a B.S in Chemistry (1975) and an M.S. in Computing Sciences (1981) from the University of California, Davis where he currently works as Data Communications Manager responsible for networking on the UC Davis campus. He also represents UC Davis as a founding member in the Bay Area Regional Research Network (BARRNet). He formed and now chairs the California Internet Federation, a forum for coordinating educational and research networks in California. In addition he is Area Director for Applications in the Internet Engineering Task Force and a member of the Internet Engineering Steering Group.

[Ed.: Interoperability demonstrations of PPP will take place on the show network during INTEROP 90, see the following page.]

Special Demonstrations for INTEROP® 90

Plans for INTEROP 90 (October 8–12, San Jose Convention Center) are already well under way. As in previous years, the show will include demonstrations of some of the latest networking hardware and software. Last year, vendors worked hard to showcase such technologies as FDDI, SNMP, OSI, X-11, NetBIOS over OSI, and the like. The opportunity to see the many different product families interoperating in a multivendor environment has always proved to be very attractive to both buyers and engineers. (In polls taken after INTEROP 89, attendees cited the technical demonstrations as one of the most important reasons to attend the exhibition.)

The year's demos

So far, the majority of the show's exhibitors have expressed interest in participating in this unique program in 1990—indeed, several groups are already well into planning their presentations. In addition to an expanded SNMP demo, this year attendees will be able to see the latest implementation of FDDI, some X.500 products, and several implementations of PPP. There are even plans underway for ISDN and SMDS demonstrations.

Planning meetings

To assist the vendors with their planning, Interop, Inc. is hosting some planning meetings concurrent with their *Internetworking Tutorials* program in the coming months. The first meetings were held in Los Angeles in January; the next meetings will be held in Boston (starting April 24th), and thereafter in Dallas in June. Since the show is just six months away, firms with an interest in demonstrating products in these areas of networking should call Chris Lynch at Interop, Inc., 415-941-3399 to find out how to get involved.

RFCs in PostScript and ASCII

Background

RFCs have been traditionally published in ASCII text. The Internet Activities Board (IAB) has decided that RFCs may be published in *PostScript*. This decision is motivated by the desire to include diagrams, drawings, and such in RFCs. It also allows authors that normally work with document production tools that produce PostScript output to use their normal tools. PostScript documents (on paper, so far) are visually more appealing and have improved readability. PostScript was chosen for the fancy form of RFC publication over other possible systems because of the perceived widespread availability of PostScript capable printers.

The need for an ASCII version

It has been pointed out that many RFC users read the documents online and use various text oriented tools (e.g., *emacs*, *grep*) to search them. Often, brief excerpts from RFCs are included in e-mail. These practices are not yet practical with PostScript files. Therefore, the IAB has also decided that whenever an RFC is published in PostScript a secondary version of that RFC is to be made available in ASCII text. This secondary version may be missing some elements of the primary version (e.g., diagrams), and be formatted differently.

Problems with PostScript

It has also been pointed out that PostScript is less standard than has been assumed and that several of the document production systems that claim to produce PostScript actually yield nonstandard results. It may be necessary to identify a set of document production systems authorized for use in production of PostScript RFCs, based on the reasonableness of the output files they generate. —*Jon Postel*

A Brief History of the Internet Engineering Task Force and the Steering Group

by Greg Vaudreuil, Corp. for National Research Initiatives

IETF

The *Internet Engineering Task Force* (IETF) is a large open community of network designers, operators, vendors, and researchers concerned with the smooth operation of the Internet and evolution of the Internet protocol architecture. The IETF began in January 1986 as a forum for technical coordination by contractors working on the ARPANET. It has grown into the primary focus for the evolution of the TCP/IP protocol suite and the management of the global Internet. The IETF mission includes:

- Specifying the short and mid term Internet protocols and architecture for the Internet Activities Board (IAB),
- Making recommendations regarding Internet Protocol Standards for IAB approval,
- Identifying and solving pressing operational and technical problems in the Internet,
- Facilitating technology transfer from the Internet Research Task Force, and
- Providing a forum for the exchange of information within the Internet community between vendors, users, agency contractors, and network managers.

History

The last 4 years have seen great progress and explosive growth. As late as the July 1986 meeting at Merit, the IETF was still essentially a 25 person group of government contractors. There was very little vendor involvement. There were no working groups and each meeting tended to be composed of network status reports and technical reports by contractors.

The first working groups did not appear until the meeting at Ames Research Center in February 1987. That meeting began a continuing focus on network management in the Internet. At the Ames meeting, Van Jacobsen presented his analysis of how TCP/IP was self-clocking in the Internet. He also presented work that would later blossom into the TCP slow-start algorithm and improved round-trip time estimation.

At the next few meetings, the IETF began to hit a technical stride with the formation of the Host Requirements, Open Shortest Path First (OSPF) routing protocol, and the Point to Point Protocol (PPP) working groups. The Host Requirements effort was meant to examine conventional wisdom and codify current practices in the implementation of host protocols. This effort wove together the separate specifications on host protocols, with the goal of defining a standard for host interoperability.

Although the IETF began with a concentration on TCP/IP, multi-protocol interoperability has also become an important goal. OSI addressing schemes for the Internet were discussed at the first meeting in January 1986. In April 1987 at BBN, the IETF had a joint meeting with X3S3.3, the ANSI group dealing with OSI network and transport layer issues. The OSI Interoperability working group had its first meeting at the January 1989 IETF at the University of Texas.

Currently the IETF plenary meetings last 3 1/2 days with 5 half day working group sessions, a day of technical briefings, network status reports, and working group progress reports. Regular reports from the operators of the major national backbones, like ARPANET, NSFnet, ESnet and the NASA Science Internet, along with reports from other network operations groups became a regular feature on the IETF plenary agenda.

IESG

As both the number of working groups and the number of attendees began to soar, it became clear that additional structure was needed to assist the IETF chair in providing technical and managerial leadership. This growth lead to the creation of the *Internet Engineering Task Force Steering Group* (IESG).

Area Directors

The IESG has the general responsibility for making the Internet operate smoothly by identifying and resolving the short and mid-term issues. The IETF was reorganized into eight technical areas, each of which is led by a technical *area director*. These eight technical directors with the chair of the IETF compose the IESG:

IETF & IESG Chair:	Phill Gross/ NRI
Applications:	Russ Hobby/ UC Davis
Host & User Services:	Craig Partridge/ BBN
Internet Services:	Noel Chiappa/ Consultant to Proteon
Routing:	Robert Hinden/ BBN
Network Management:	Dave Crocker/ DEC
OSI Integration:	Rob Hagens/ U Wisc. & Ross Callon/ DEC
Operations:	Phill Gross/ NRI (interim)
Security:	Steve Crocker/ TIS

The *Applications Area* chaired by Russ Hobby is focused on increasing the utility of the Internet. By defining and encouraging the use of new applications this effort will make the Internet environment easier to use.

The *Host and User Services Area* chaired by Craig Partridge is responsible for the evolution of host based services at the transport level and above, including dynamic configuration and TCP enhancements. This area is also concerned with improving the quality of user services by documenting existing resources and developing new tools, protocols, and procedures to help users make better use of the Internet.

The *Internet Services Area* chaired by Noel Chiappa is primarily concerned with the evolution of the IP protocol and ensuring interoperability across the Internet. This effort involves the development of protocols for transmission of IP over new physical media, common requirements for Internet routers, and enhancements to the IP level protocols.

As the Internet grows, Bob Hinden as chair of the *Routing Area*, coordinates the development of new routing protocols and the interoperability of existing protocols. Current efforts in this area include selecting a standard inter gateway protocol (IGP) and the development of a new exterior gateway protocol (EGP).

Network Management has proven to be an all-encompassing effort. Dave Crocker is working for a unified architecture for network management which will allow common interfaces for network monitoring and traffic accounting.

continued on next page

A Brief History of the IETF and the IESG *(continued)*

This effort is focused on defining common Management Information Base (MIB) variables "The MIB" for new and existing network technologies and protocols for collecting and exchanging MIB information.

As the Internet world evolves toward a multi-protocol environment, Ross Callon and Rob Hagens (as co-chairmen of the *OSI Integration Area*) concentrate on providing the framework for incorporating OSI protocols and services into the Internet environment. In recent months, their work has focused upon the OSI network layer (including OSI routing), X.400 and X.500.

The *Operations Area*, chaired for the interim by Phill Gross, is concerned with the operational stability of the Internet. Efforts in this area involve managing the growth of the Internet, both by engineering a sane topology and providing operational guidelines for participants in the Internet.

The *Security Area*, chaired by Steve Crocker, is focused on development of a security policy and security architecture for the Internet. This work includes extension of existing protocols to include confidentiality, integrity and authentication mechanisms, and protection of the network infrastructure from disruption of service. This area is closely coordinated with the *Privacy and Security Research Group* and with the other IETF areas.

Participation in the IETF

There is no formal membership in the IETF. The work is done by individuals who share an interest in the resolution of particular problems. The work of the IETF is conducted in Working Groups, each of which is convened to solve a particular problem, work on an enhancement or exchange information vital to the operation of the Internet. There are currently over 40 working groups. The working groups conduct business via electronic mailing lists established for each group and during plenary meetings of the IETF. The IETF holds quarterly plenary sessions composed of working group sessions, technical presentations and network status briefings.

Meetings

The next two plenary sessions are May 1–4, 1990 at the Pittsburgh Supercomputer Center, and July 31–August 3, 1990 at the University of British Columbia. Information and logistics about upcoming meetings of the IETF are distributed on the IETF mailing list. To join the list or for inquiries about the IETF, send a request to `ietf-request@isi.edu`.

More information

Charters, meeting reports, and other IETF information is available online for anonymous FTP from the IETF directory at `nic.ddn.mil` and `nnsc.nsf.net`. Proceedings of the IETF quarterly plenary meetings are produced and are available from the Corporation for National Research Initiatives (NRI). You may order a copy by sending a check for $35 to IETF Proceedings, NRI, 1895 Preston White Drive, Suite 100, Reston, VA 22091. Additionally, information about the current working groups, their charters and meeting reports are kept on line in the IETF directories at `nnsc.nsf.net` and `nic.ddn.mil`.

GREG VAUDREUIL graduated from Duke University with a degree in Electrical Engineering and a major in Public Policy Studies. He was thrust into the heart of the IETF by accepting a position with the Corporation for National Research Initiatives to manage the explosive growth of the IETF.

Network Management Tool Catalog now available

by Bob Stine, SPARTA, Inc.

Introduction

The Internet Engineering Task Force (IETF) has recently completed a catalog of tools for managing TCP/IP internets and the communications resources that they interconnect. The document is called *A Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices*, and is published as RFC 1147.

Content

The catalog has brief descriptions of 85 network management tools. Each entry tells what a tool does, how it works, and how it can be obtained. The catalog also includes an introductory tutorial on the practice of network management. [Ed.: An adapted version of this network management tutorial will appear in a future issue of *ConneXions*.]

The goal of the catalog is to provide practical information to site administrators and network managers. It identifies tools that are available to assist network managers in debugging and maintaining their networks. Network management tools and LAN management tools (other than breakout boxes) are listed. Public domain, copyrighted, and commercial tools are listed.

Updates

Future editions of the catalog will be issued as IETF members become aware of tools that should be included, or of deficiencies or inaccuracies in the current edition. Comments, updates, and new tool descriptions are welcome, and should be sent to Robert Stine, by email, to `stine@sparta.com`, or by US Mail, to:

R.H. Stine
SPARTA, Inc.
7926 Jones Branch Drive, Suite 1070
McLean, VA 22102.

Getting RFCs

RFCs can be obtained via FTP from `nic.ddn.mil`, with the pathname RFC:RFCnnnn.TXT (where "nnnn" refers to the number of the RFC). Login with FTP, username ANONYMOUS and password GUEST. The NIC also provides an automatic mail service for those sites which cannot use FTP. Address the request to: `Service@nic.ddn.mil` and in the subject field of the message indicate the RFC number, as in "`Subject: RFC 1147`." Contact the NIC for information on obtaining hardcopy versions of RFCs. Their telephone numbers are: 1-800-235-3155 or 1-415-859-3695. The NIC is located at:

DDN Network Information Center
SRI International
333 Ravenswood Avenue
Menlo Park
California 94025

Requests for special distribution should be addressed to either the author of the RFC in question, or to `nic@nic.ddn.mil`. Unless specifically noted otherwise on the RFC itself, all RFCs are for unlimited distribution. Submissions for Requests for Comments should be sent to: `postel@isi.edu`.

Upcoming Events

IFIP workshop on high-speed networking

The *Second International Workshop on Protocols for High-Speed Networks* will be held November 27–29, 1990 in Palo Alto, California. The workshop is Sponsored by IFIP WG6.1/WG6.4. The workshop will focus on problems of obtaining high throughput and fast response time over either LANs or WANs.

Call for papers

Original research papers are solicited in the following areas:

- Protocol development
- Protocol analysis
- High-performance architectures
- Implementation of high-speed protocols
- Experimental studies
- Application-oriented issues

Papers on related topics will also be considered.

Important dates

Notification of intent to submit paper:	Immediately
Submission of paper (3 copies):	June 15, 1990
Notification of acceptance:	September 15, 1990
Camera-ready paper for proceedings due:	October 22, 1990

Send Papers To:

Marjory Johnson, Chairperson
RIACS, Mail Stop 230-5
NASA Ames Research Center
Moffett Field, California 94035
USA
E-mail: mjj@riacs.edu
Telephone: 415-604-6363
FAX: 415-961-8467

Publication

Selected papers will be published either as a North-Holland book or in a special issue of *Computer Networks and ISDN Systems*.

Venue

The workshop will be held at the Holiday Inn, Palo Alto, adjacent to the Stanford University campus. Registration fee will be approximately $200 (U.S.).

SIGCOMM '90

ACM SIGCOMM '90 will be held in Philadelphia, Pennsylvania, September 25–27th, with tutorials on September 24th. Registration information will appear in the July issue of *Computer Communication Review* and the August issue of *IEEE Communications*. For more information, contact the General chair:

David Farber,
Professor of Computer and Information Science
and of Electrical Engineering
University of Pennsylvania
200 South 33rd Street
Philadelphia, Pennsylvania, 19104-6389
Telephone: 215-898-9508 FAX: 215-898-0587
E-mail: farber@cis.upenn.edu

Announcing The INTEROP Achievement Award

Background

At every INTEROP conference and exhibition you have the opportunity to hear about the latest developments in the field of multivendor interoperability and see demonstrations of networking technology on the exhibit floor. Researchers will tell you what's going on with protocol development, and vendors will show you their newest networking products. However, the most impressive achievements in this field are neither found in research laboratories nor manifested by product announcements. The real test of the technology takes place *in the field*, at thousands of sites all over the world. The operators of these sites are the people who actually *use* interoperable systems.

Four categories

In order to honor organizations that make the most effective use of internetworking technology, Interop Inc. is proud to announce the *INTEROP Achievement Award*. The award will be presented for the first time during INTEROP 90 to organizations in each of the following business sectors:

- Manufacturing
- Service
- Government
- Education

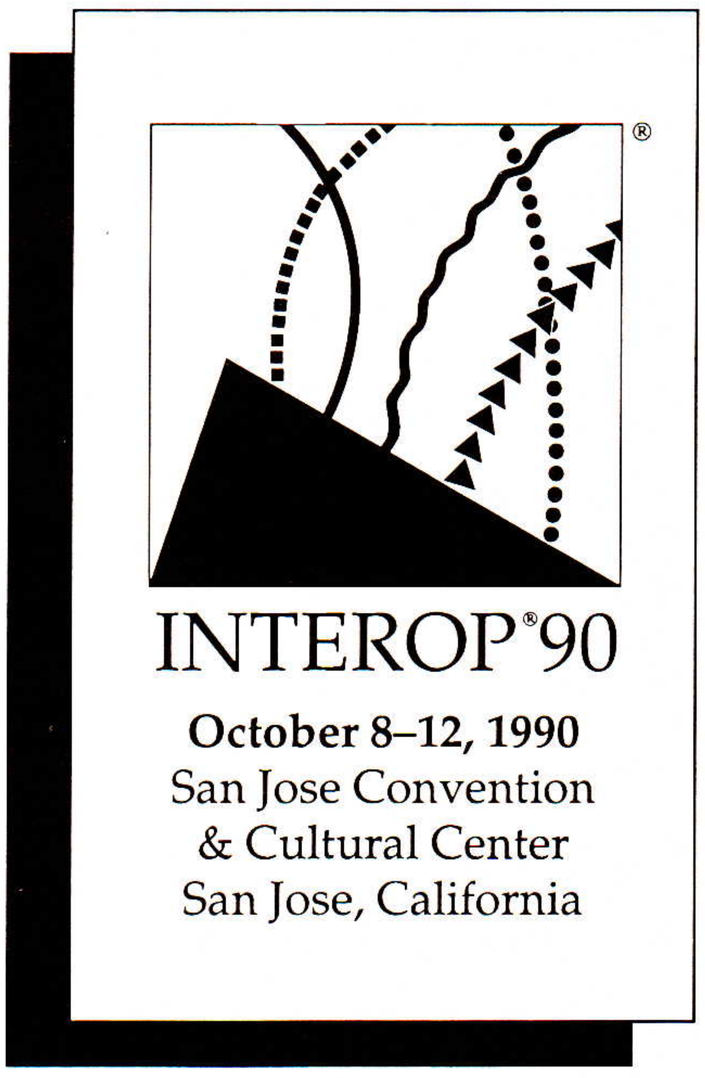
Likely candidates

Winners will have demonstrated a successful use of existing technologies and applications to accomplish their goals. They most likely will be using TCP/IP and OSI, and probably will have had to incorporate proprietary technologies. A mixture of LAN and WAN technologies is likely. Some organizations might even have international locations on their internets, with the increased complexity that such links represent. We are looking for organizations that are "pushing the envelope" in order to win in their business sector.

Nominations

Nominations for the award will be judged by a distinguished panel of communications industry experts, including vendors and members of the press. All entries must be received by June 1, 1990 and must be submitted on a special nomination form. The form is available from:

Interop, Inc.
480 San Antonio Road
Suite 100
Mountain View
California, 94040
USA
Attention: Wendy Gibson
Telephone: 1-415-941-3399
FAX: 1-415-949-1779
Toll-free: 1-800-INTEROP



CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen